

**(12) NACH DEM VERTRÄG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES  
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG**

## **(19) Weltorganisation für geistiges Eigentum Internationales Büro**



**(43) Internationales Veröffentlichungsdatum**  
**24. Januar 2002 (24.01.2002)**

PCT

(10) Internationale Veröffentlichungsnummer  
**WO 02/06950 A2**

**(51) Internationale Patentklassifikation<sup>7</sup>:** G06F 9/00

G06F 9/00

(71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von US): SIEMENS AKTIENGESELLSCHAFT [DE/DE]; Wittelsbacherplatz 2, 80333 München (DE).

(21) Internationales Aktenzeichen: PCT/DE01/02612

### (72) Erfinder; und

(75) **Erfinder/Anmelder (nur für US): BERENTROTH, Lutz [DE/DE]; Cecilia-Vogl-Weg 36A, 93055 Burgweinting (DE). HÖLZL, Stefan [DE/DE]; Hadamarstrasse 28a, 93051 Regensburg (DE). WELLNHOFER, Helmut [DE/DE]; Erich-Ollenhauer-Strasse 24, 93077 Bad Abbach (DE).**

(25) Einreichungssprache: Deutsch

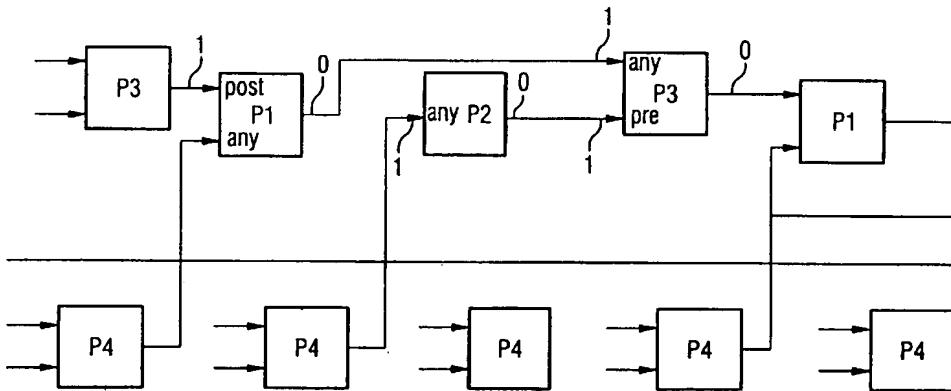
(74) Gemeinsamer Vertreter: SIEMENS AKTIENGESELLSCHAFT; Postfach 22 16 34, 80506 München (DE).

(30) Angaben zur Priorität:  
100 34 869.6 18. Juli 2000 (18.07.2000) DE

[Fortsetzung auf der nächsten Seite]

(54) Title: METHOD FOR AUTOMATICALLY OBTAINING AN OPERATIONAL SEQUENCE OF PROCESSES AND A CORRESPONDING TOOL THEREFOR

**(54) Bezeichnung: VERFAHREN ZUM AUTOMATISCHEN GEWINNEN EINER FUNKTIONSFÄHIGEN REIHENFOLGE VON PROZESSEN UND WERKZEUG HIERZU**



WO 02/06950 A2

**(57) Abstract:** In order to automatically calculate an operational sequence of processes that determine an output value from at least one input value, a multitude of processes (P1 - P8), whose inputs are provided with at least one of the attributes: input value of the same calculation cycle (PRE), input value of the preceding calculation cycle (POST), input value from any calculation cycle (ANY), are arranged in such a manner that a process, which does not have any input with the attribute input value of the same calculation cycle (PRE), is determined as the first process of a calculation cycle and, in successive analogous steps, determines a quantity of possible sequences.

**(57) Zusammenfassung:** Zum automatischen Berechnen einer funktionsfähigen Reihenfolge von Prozessen, die aus wenigstens einem Eingabewert einen Ausgabewert bestimmen, werden eine Vielzahl von Prozessen (P1 - P8), deren Eingänge mit wenigstens einem der Attribute: "Eingangswert des selben Berechnungszyklus" (PRE), "Eingangswert des vorhergehenden Berechnungszyklus" (POST), "Eingangswert aus beliebigem Berechnungszyklus" (ANY) versehen sind, so angeordnet, dass als erster Prozess eines Berechnungszyklus ein Prozess bestimmt wird, der keinen Eingang mit dem Attribut "Eingangswerts des selben Berechnungszyklus" (PRE) aufweist, und in nachfolgenden analogen Schritten eine Menge von möglichen Reihenfolgen bestimmt.





(19) BUNDESREPUBLIK

DEUTSCHLAND



DEUTSCHES

PATENT- UND

MARKEANAMT

(12) **Offenlegungsschrift**  
**DE 100 34 869 A 1**

(51) Int. Cl. 7:  
**G 05 B 19/04**

(21) Aktenzeichen: 100 34 869.6  
(22) Anmeldetag: 18. 7. 2000  
(23) Offenlegungstag: 7. 2. 2002

(71) Anmelder:  
Siemens AG, 80333 München, DE

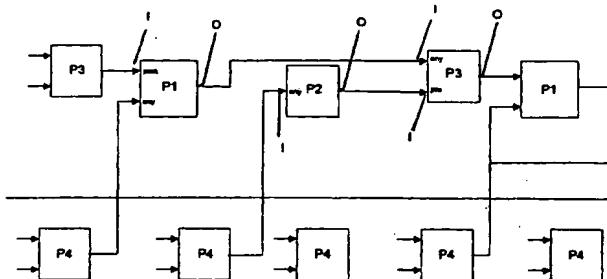
(72) Erfinder:  
Berentroth, Lutz, 93055 Regensburg, DE; Hölzl, Stefan, 93051 Regensburg, DE; Weinhöfer, Helmut, 93077 Bad Abbach, DE

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen**

Prüfungsantrag gem. § 44 PatG ist gestellt

(54) Verfahren zum automatischen Gewinnen einer funktionsfähigen Reihenfolge von Prozessen und Werkzeug hierzu

(57) Zum automatischen Berechnen einer funktionsfähigen Reihenfolge von Prozessen, die aus wenigstens einem Eingabewert einen Ausgabewert bestimmen, werden eine Vielzahl von Prozessen (P1-P8), deren Eingänge mit wenigstens einem der Attribute  
- "Eingangswert desselben Berechnungszyklus" (PRE),  
- "Eingangswert des vorhergehenden Berechnungszyklus" (POST),  
- "Eingangswert aus beliebigem Berechnungszyklus" \* (ANY) versehen sind, so angeordnet, dass als erster Prozess eines Berechnungszyklus ein Prozess bestimmt wird, der keinen Eingang mit dem Attribut "Eingangswerts desselben Berechnungszyklus" (PRE) aufweist, und in nachfolgenden analogen Schritten eine Menge von möglichen Reihenfolgen bestimmt.



DE 100 34 869 A 1

DE 100 34 869 A 1

## Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zum automatischen Gewinnen einer funktionsfähigen Reihenfolge von Prozessen, ein Verfahren zum Überprüfen einer Reihenfolge von abzuarbeitenden Prozessen auf Inkonsistenzen im Datenfluß und ein Werkzeug zum Anordnen oder Prüfen einer Reihenfolge von abzuarbeitenden Prozessen, insbesondere einer Sequenz von Berechnungsfunktionen einer Betriebssystemsoftware, beispielsweise einer Betriebssystemsoftware eines elektronischen Steuergeräts für ein Kraftfahrzeug.

[0002] Aus der Veröffentlichung "Kürzere Entwicklungszeiten mit automatischer Codegenerierung" in Automotive Engineering Partners 5/99, Vieweg GmbH, Verlag Friedrich & Sohn Verlagsgesellschaft mbH in den GWV Fachverlagen GmbH, ist ein Werkzeug zur automatischen Code-Generierung für Betriebssystemsoftware von Kraftfahrzeug-Steuergeräten bekannt sowie ein Testwerkzeug zum Überprüfen der Funktionen eines Steuergeräts. Das Werkzeug zur automatischen Codegenerierung übersetzt automatisch die technischen Vorgaben von Entwicklungingenieuren, die in Form einer physikalischen Beschreibung einer Regelung vorliegen, in den für ein Steuergerät eines Kraftfahrzeugs lesbaren Code. Das Testwerkzeug spricht das Steuergerät über eine Schnittstelle an und erzeugt physikalisch korrekte Signale, die im realen Betrieb eines Kraftfahrzeugs erzeugt würden. Es kann daher das Systemverhalten des Steuergeräts, und genauer dessen Betriebssystemsoftware, in einer Simulation des geschlossenen Regelkreises frühzeitig im Labor untersucht werden.

[0003] Das Werkzeug ist aber nicht in der Lage, selbständig die Reihenfolge von Prozessen einer Regelung zu bestimmen, sondern dient im wesentlichen nur zur Übersetzung von menschlichen Vorgaben in Softwarecode. Ferner ist es häufig wünschenswert, aus vorhandenen Bibliotheken bereits getestete Reihenfolgen von Prozessen oder einzelne Prozesse für eine Neuentwicklung zu übernehmen und neue Prozesse an einer geeigneten Stelle der vorhandenen Reihenfolge einzufügen beziehungsweise bekannte, geprüfte Prozesse mit neu zu erstellenden Prozessen zu einer lauffähigen Sequenz zu verbinden.

[0004] Es ist ein Ziel der Erfindung, ein Verfahren zum automatischen Berechnen einer funktionsfähigen Reihenfolge von Prozessen, ein Verfahren zum Überprüfen einer Reihenfolge von abzuarbeitenden Prozessen und ein Werkzeug bereitzustellen, die in der Lage sind eine Vielzahl von Prozessen so zu ordnen, dass eine zielgerichtete Abarbeitung aller Prozesse erfolgen kann, die dem gewünschten Datenfluß genügt.

[0005] Dieses Ziel wird mit den Verfahren und einem Werkzeug erreicht, wie sie in den unabhängigen Ansprüchen definiert sind. Vorteilhafte Ausführungsformen der Erfindung sind Gegenstand der Unteransprüche.

[0006] Durch die Zuordnung eines der Attribute "Eingangswert des selben Berechnungszyklus", "Eingangswert des vorhergehenden Berechnungszyklus" oder "Eingangswert aus beliebigem Berechnungszyklus" zu jedem Dateneingang der zu ordnenden Prozesse ist es möglich, maschinel eine funktionsfähige, gültige Reihenfolge oder Sequenz zu schaffen, die keine Inkonsistenzen im Datenfluß aufweist. Eine Inkonsistenz bedeutet, dass die den Eingängen der Prozesse zugeordneten Attribute logisch nicht erfüllt werden können, da sie sich gegenseitig ausschließen. Beispielsweise können die Eingänge zweier Prozesse, die gegenseitig voneinander abhängen, nicht jeweils mit dem Attribut "PRE" versehen sein, da sonst die widersprüchliche Bedingung erfüllt sein müsste, dass jeder der beiden Pro-

zesse vor dem anderen Prozess berechnet werden müsste. In der Regel gibt es aber eine Vielzahl von "lauffähigen" Reihenfolgen.

[0007] Aus der Menge von lauffähigen Reihenfolgen von Prozessen kann diejenige bestimmt werden, die die geringste Laufzeit aufweist oder diejenige, bei der die Prozesse, deren Ausgangswerte in nachfolgenden Prozessen als Eingangsvere verwendet werden, am dichtensten aufeinander folgen (Optimierung).

[0008] Außerdem können innerhalb eines Berechnungszyklus Prozesse identifiziert werden, die keine Abhängigkeiten voneinander aufweisen. Solche voneinander unabhängigen Prozesse können in Recheneinheiten mit Multitasking-Betriebssystemen, insbesondere in Mehrprozessorsystemen, parallel abgearbeitet werden.

[0009] Eine Inkonsistenz in einer Reihenfolge von Prozessen kann beispielsweise folgendermaßen ermittelt werden:

## 1. Schritt

[0010] Bestimmung eines Prozesses, der keinen Eingang mit dem Attribut "Eingangswert des selben Berechnungszyklus" hat (bezogen auf die aktuelle Menge der Prozesse).

## 2. Schritt

[0011] Entfernen dieses Prozesses aus der Menge der Prozesse.

## 3. Schritt

[0012] Wiederholen der Schritte 1 und 2 bezüglich der restlichen Menge von Prozessen.

[0013] Lässt sich die Menge der Prozesse durch obigen Vorgang nicht reduzieren, bis kein Prozess übrig ist, ist eine Inkonsistenz gefunden.

[0014] Es ist daher möglich, die Reihenfolge der Prozesse als lauffähig zu bestätigen, ohne sie in einem realen System mit einer aufwendigen Simulation des gesamten Systemverhaltens testen zu müssen.

[0015] Die vorliegende Erfindung kann beispielsweise zum Entwerfen von Betriebssystemsoftware für Computer, von komplexen Anwendungsprogrammen, von Ablaufsteuerungen für eine industrielle Fertigung und insbesondere zum Bestimmen von Steuerungen für Steuergeräte in der Kraftfahrzeugtechnik eingesetzt werden, beispielsweise von Motor- und Getriebesteuergeräten.

[0016] Weitere Merkmale, Vorteile und Anwendungsmöglichkeiten der Erfindung ergeben sich aus der nachfolgenden Beschreibung von Ausführungsbeispielen in Verbindung mit den Zeichnungen. Es zeigen:

[0017] Fig. 1 ein Werkzeug zum Anordnen oder Prüfen,

[0018] Fig. 2 die Verknüpfung von Prozessen innerhalb einer Task,

[0019] Fig. 3 die Prozesse einer Drehmoment-Regelung einer Motorsteuerung,

[0020] Fig. 4 die Darstellung der Verknüpfung der Prozesse von Fig. 3 in Form eines Graphen,

[0021] Fig. 5-8 die Auflösung des Graphen von Fig. 4,

[0022] Fig. 9 die Zuordnung der Prozesse der Fig. 3 zu sequentiell aufeinander folgenden Prozessgruppen, und

[0023] Fig. 10-13 die Überprüfung einer Reihenfolge von Prozessen, die eine Inkonsistenz aufweisen.

[0024] Fig. 1 zeigt ein Werkzeug zum Anordnen und/oder

Prüfen einer Reihenfolge von abzuarbeitenden Prozessen. Dieses weist eine Recheneinheit 1 mit einem Mikroprozessor MP und einem Arbeitsspeicher RAM auf. Mit der Recheneinheit 1 sind als Eingabemittel ein Laufwerk 2 für ein

Speichermedium 21, bei dem es sich um eine beschreibbare CD (Compact Disk) handelt, und eine Tastatur 3 vorgesehen.

[0025] Auf dem Speichermedium 21 sind Reihenfolgen von Prozessen für die Ausführung von Aufgaben (Tasks) einer Betriebssystem-Software oder einer Anwendungssoftware und einzelne Prozesse gespeichert. Es handelt sich also um eine Bibliothek mit bereits vorhandenen Prozessen, die für die Erstellung von neuen Programmen benutzt werden kann.

[0026] Mit der Tastatur 3 können ebenfalls Prozesse in die Recheneinheit 1 eingegeben werden.

[0027] Zur Darstellung von Berechnungsergebnissen der Recheneinheit 1 dient ein Ausgabemittel, bei dem es sich um eine Anzeigeeinrichtung 4 handelt.

[0028] Fig. 2 veranschaulicht die miteinander verknüpften Prozesse P1-P3 eines Berechnungszyklus n, die eine Task oder Aufgabe eines Betriebssystems einer Recheneinheit oder eines Anwendungsprogramms abarbeiten. Die Prozesse P1-P3 sind Prozesse, die regelmäßig wiederkehrend abgearbeitet werden. Der Prozess 4 ist ein asynchroner Prozess, der bei bestimmten Ereignissen aufgerufen wird, Daten für Prozesse innerhalb des Berechnungszyklus liefert aber nicht zum betrachteten Berechnungszyklus gehört.

[0029] Eine Task ist eine geordnete Reihenfolge oder Sequenz von Prozessen, die vom Betriebssystem oder einem Programm aufgerufen wird. Der Aufruf kann zyklisch in regelmäßigen Intervallen oder asynchron aufgrund von bestimmten Ereignissen erfolgen. Ein Prozess ist eine Gruppe von Anweisungen, die als Einheit eine bestimmte Aufgabe erfüllen. Prozesse konsumieren Eingangsdaten und erzeugen Ausgabedaten. Ein Prozess ist eine Einheit von Anweisungen, die durch Interrupts unterbrochen werden kann, davon abgesehen aber als Einheit abgearbeitet wird. Aufgrund der Abhängigkeit der erzeugten von den konsumierten Daten ist die Sequenz oder Reihenfolge der Prozesse für das Berechnungsergebnis entscheidend.

[0030] Die Sequenz der Prozesse P1-P3 ist implizit durch die Verwendung von zusätzlichen Attributen für den Typ eines Eingangs oder Eingangsports I definiert. Das Attribut bzw. die Attribute eines Eingangs I legt bzw. legen die relative Reihenfolge eines Prozesses zu den anderen Prozessen des selben Berechnungszyklus n fest. Es gibt drei verschiedene Typen von Eingangsports:

- Eingangswert des selben Berechnungszyklus: PRE,
- Eingangswert des vorherigen Berechnungszyklus: POST,
- Eingangswert aus beliebigem Berechnungszyklus: ANY.

[0031] Im folgenden werden die Typen der Eingänge oder Eingangsports regelmäßig durch ihre Bezugszeichen PRE, POST und ANY bezeichnet.

[0032] Ein Eingang I des Typs PRE benötigt einen Eingangswert, der innerhalb des selben Berechnungszyklus bereits zur Verfügung gestellt worden ist, also vorhergehend erzeugt worden ist. Der Prozess, dessen Ausgang beziehungsweise Ausgabeport O den Wert für den mit dem Attribut PRE versehenen Eingang I liefert, muss also in der Reihenfolge vorher berechnet worden sein.

[0033] Prozess P3 weist einen Eingang I des Typs PRE auf. Dieser Eingang I konsumiert einen Wert, der vom Ausgang O des Prozesses P2 zur Verfügung gestellt wird. Das Werkzeug zum Anordnen oder Prüfen einer Reihenfolge von Prozessen erkennt daher, dass Prozess P2 vor Prozess P3 in die Sequenz eingereicht werden muss.

[0034] Bei einem Eingang I des Typs POST muß der be-

reitgestellte Wert erst später in der Sequenz erzeugt werden, so dass beim Abarbeiten des Berechnungszyklus n ein alter, im vorhergehenden Berechnungszyklus n - 1 berechneter Eingabewert zur Verfügung steht (dies entspricht einem z<sup>-1</sup>-Glied in der Regelungstechnik). Das Werkzeug zum Anordnen oder Prüfen wird daher Prozess P3, dessen Ausgabewert von einem Eingang I des Typs POST vom Prozess P1 benötigt wird, nach Prozess P1 im selben Berechnungszyklus n anordnen.

[0035] Bei einem Eingang vom Typ ANY spielt es keine Rolle, ob der konsumierte Wert im selben Berechnungszyklus n berechnet wurde oder im vorhergehenden Berechnungszyklus n - 1 berechnet wurde. Die Belegung eines Eingangs I mit dem Attribut ANY ist daher lediglich optional.

Häufig erhält ein Eingang vom Typ ANY seinen Eingangswert von außerhalb der Task. In Fig. 2 ist der Prozess P4 solch ein externer Prozess. Eingangsports, die ihre Daten von externen Prozessen erhalten, haben grundsätzlich das Attribut ANY.

[0036] Aufgrund der vergebenen Attribute für die Eingänge I der Prozesse P1-P3 können Ungleichungen aufgestellt werden. Auf diese Weise wird jeweils eine Beziehung zwischen einem Prozess hergestellt, dessen Ausgangsport O einen Wert bereitstellt, und einem Prozess, dessen Eingangsport I, diesen bereitgestellten Wert konsumiert.

[0037] Liefert ein Prozess einen Eingangswert für einen Eingang I vom Typ PRE, so wird dieser Prozess kleiner als der Prozess mit diesem Eingang vom Typ PRE gesetzt. Der Prozess der in einer Relation als "kleiner" gesetzt ist, muss vor dem anderen Prozess der Relation ausgeführt werden.

Ist der Eingang I dagegen vom Typ POST, so wird der Prozess, der den Eingangswert liefert, größer als der Prozess mit diesem Eingang vom Typ POST gesetzt. Ist der Eingang vom Typ ANY, so kann der den Eingangswert liefernde Prozess größer oder kleiner als der den Eingangswert empfangende Prozess gesetzt werden. Der Eingang vom Typ ANY kann schließlich von einem im selben oder nachfolgenden Berechnungszyklus angeordneten Prozess bedient werden. Die Anordnung der beiden Prozesse zueinander ist also beliebig. Hierdurch ergeben sich so viele Ungleichungen, wie es im Berechnungszyklus Eingänge mit dem Attribut PRE oder POST gibt.

[0038] Für dieses Beispiel ergeben sich daher folgende Ungleichungen:  
 45 P3 > P1 oder P1 < P3, und  
 P2 < P3

[0039] Als erster Prozess einer Berechnungsreihenfolge oder Sequenz kommt nur ein Prozess in Frage, der keinen Eingang vom Typ PRE aufweist. Als letzter Prozess einer Sequenz kommt nur ein Prozess in Frage, der keinen Eingang vom Typ POST aufweist.

[0040] Danach kommen die Prozesse P1 und P2 als erste Prozesse der Sequenz in Frage. Prozess P3 darf erst nach den Prozessen P1 und P2 berechnet werden. Die Prozesse P1 und P2 können in beliebiger Reihenfolge oder parallel abgearbeitet werden.

[0041] Fig. 3 zeigt zur Veranschaulichung eine vereinfachte Task mit einer Berechnungsreihenfolge oder Sequenz von Prozessen P1-P8 einer Drehmoment-Regelung eines 60 Betriebssystems eines Motorsteuergeräts eines Kraftfahrzeugs. Typischerweise weist eine derartige Task allerdings eine Sequenz von 100 bis 200 Prozessen auf.

[0042] Die Prozesse sind zum Teil aus einer Bibliothek entnommen. Einige Prozesse wurden neu für die Drehmoment-Regelung entworfen. Die Grafik von Fig. 3 steht einem Entwickler, der eine gültige Sequenz von Prozessen finden will, zur Verfügung, sie bestimmt jedoch noch keine Prozessreihenfolge.

[0043] Prozess P1 berechnet für eine Benutzereinstellung einer Geschwindigkeitsregelanlage (Tempomat) die Differenz zwischen der tatsächlichen Geschwindigkeit und der von einem Fahrer eingestellten Geschwindigkeit und gibt diese Differenz an Prozess P3 aus. Prozess P2 berechnet eine durch eine Gaspedalstellung mitgeteilte Leistungsanforderung und gibt diese ebenfalls an Prozess P3 aus. Prozess P3 koordiniert eventuell vorliegende Anforderungen der Prozesse P1 und P2 und berechnet hierfür eine Drehmomentanforderung für die Prozesse P4 und P6.

[0044] Prozess P4 bestimmt eine langsame Drehmomentänderung und gibt eine entsprechende Anforderung an Prozess P5 aus. Prozess P5 berechnet einen Sollwert für eine Drosselklappenstellung und gibt diesen an ein Stellglied aus.

[0045] Prozess P6 berechnet eine schnelle Drehmomentänderung und gibt eine entsprechende Anforderung an Prozess P7 aus, der einen Sollwert für einen Zündzeitpunkt bestimmt und an die Zündung ausgibt.

[0046] Prozess P8 enthält ein Referenzsystem für die Effektivität einer Drehmomentänderung durch eine Zündzeitpunktverschiebung. Die Prozesse P4 und P6 entscheiden anhand ihrer Kopplung und der Effektivität, wie der Aufbau eines bestimmten Drehmoments auf die zwei möglichen Stellgrößen, Drosselklappe und Zündwinkel, verteilt werden muss.

[0047] Die Task wird zyklisch, z. B. alle 10 ms, vollständig berechnet. Alle Eingangswerte der Prozesse P3 bis P8, mit Ausnahme der mit "eff" gekennzeichneten Eingangswerte beziehungsweise Eingänge, müssen im selben Berechnungszyklus vorher berechnet worden sein. Sie müssen also das Attribut "Eingangswert des selben Berechnungszyklus" PRE erhalten. Die mit "eff" bezeichneten Eingangswerte müssen im vorhergehenden Berechnungszyklus berechnet worden sein, das heißt, bei der vorangegangenen Ausführung der Task. Dementsprechend erhalten diese Eingangswerte beziehungsweise die entsprechenden Eingänge das Attribut "Eingangswert des vorhergehenden Berechnungszyklus" POST.

[0048] Für jeden Eingang der Prozesse P1-P8 kann also angegeben werden, ob ein aktuell im selben Berechnungszyklus berechneter Wert oder ein im vorhergehenden Berechnungszyklus bei der vorangehenden Abarbeitung der Task ermittelter Wert benutzt werden muss. Letzteres entspricht einem  $z^{-1}$ -Glied aus der Regelungstechnik. Die Eingangswerte der Prozesse P1 und P2 sind nicht definiert. Sie können also das Attribut "Eingangswert aus beliebigem Berechnungszyklus" ANY erhalten.

[0049] Aufgrund dieser Prämissen ergeben sich für die Prozesse P1-P8 folgende Relationen:

P1 < P3  
P2 < P3  
P3 < P4  
P3 < P6  
P4 < P5  
P6 < P4  
P4 < P8  
P6 < P7  
P6 < P8  
P7 < P8

[0050] Als erste Prozesse der Sequenz kommen lediglich Prozesse in Frage, die keinen im selben Berechnungszyklus berechneten Eingangswert benötigen. Der erste Prozess darf also keinen mit dem Attribut PRE belegten Eingang aufweisen. Der letzte Prozess der Sequenz darf keinen mit dem Attribut POST belegten Eingang aufweisen. Alle Berechnungsreihenfolgen die diesen Bedingungen und den aufgestellten Ungleichungen genügen, sind gültige oder lauffähige Sequenzen.

[0051] Das Ungleichungssystem kann beispielsweise durch Anwendung der Graphentheorie computergestützt gelöst werden. Ein Graph wird aufgebaut, indem die Prozesse

5 als Knoten und jede Ungleichung als gerichtete Kante des Graphen eingesetzt werden. Eine gerichtete Kante ist ein Verweis von einem Prozess auf einen anderen Prozess. Die gerichtete Kante wird jeweils von dem Prozess bzw. Knoten, der in der entsprechenden Ungleichung als kleiner bestimmt 10 wurde, auf den Prozess bzw. Knoten gerichtet, der in der selben Ungleichung als größer bestimmt wurde.

[0052] Anders ausgedrückt wird jeweils zu einem Prozess, dessen Eingang mit dem Attribut PRE versehen ist, ein Verweis von dem Prozess gesetzt, der den Eingangswert liefert. 15 Von einem Prozess, dessen Eingangswert mit dem Attribut POST versehen ist, wird ein Verweis auf den Prozess gesetzt, der den Eingangswert liefert.

[0053] In Fig. 4 ist die Umsetzung der oben aufgestellten Ungleichungen in einen Graphen dargestellt. In einem ersten Bearbeitungsschritt oder einer ersten Iteration werden alle Knoten (Prozesse) aus der Menge der zu überprüfenden Knoten (Prozessen) entfernt, einschließlich der zugehörigen Verweise E (Kanten), zu denen keine Verweise E (Kanten) des Graphen zeigen. Dies gilt für die Prozesse P1 und P2, 20 die keinen Eingang mit dem Attribut PRE aufweisen. Diese Prozesse werden einer ersten Gruppe von Prozessen des Berechnungszyklus zugeordnet.

[0054] Die Prozesse innerhalb einer Gruppe von Prozessen können in beliebiger Reihenfolge oder parallel abgearbeitet werden.

[0055] Durch rekursive Abarbeitung des Algorithmus wird jeweils eine neue Gruppe von Prozessen gebildet, die bei der Programmausführung nach der vorhergehenden Gruppe von Prozessen abgearbeitet werden muss. Es werden also die nach jedem Schritt verbleibenden Prozesse in weiteren aufeinanderfolgenden Schritten jeweils auf Prozesse untersucht, auf die kein Verweis E gesetzt ist. Die in jedem Schritt gefundenen Prozesse werden entfernt und aufeinander folgenden Gruppen von Prozessen zugeordnet.

[0056] 25 Diese Gruppen können einen oder mehrere Prozesse beinhalten. Die Reihenfolge der Prozesse innerhalb einer Gruppe ist beliebig, da diese Prozesse voneinander unabhängig sind. Es müssen jedoch alle Prozesse einer Gruppe abgearbeitet sein, bevor ein Prozess der nächsten Gruppe 30 ausgeführt wird.

[0057] 35 In Fig. 5 ist der um die Prozesse P1 und P2 verminderte Graph gezeigt. Auch in diesem Teilgraphen muss mindestens ein Knoten (Prozess) existieren, zu dem keine Kante zeigt. Alle Knoten, zu denen keine Kante zeigt, müssen wiederum entfernt werden. Existiert kein Knoten (Prozess) der diese Bedingung erfüllt, ist eine nicht auflösbare Schleife und damit eine Inkonsistenz im Datenfluss entdeckt.

[0058] 40 In einem zweiten Schritt wird folglich der Prozess P3 entfernt und einer zweiten Gruppe von Prozessen zugeordnet.

[0059] 45 Das Resultat ist in Fig. 6 dargestellt.

[0060] 50 In einem dritten Schritt wird nun Prozess 6 entfernt, zu dem keine Kante zeigt.

[0061] 55 Der so entstandene restliche Graph ist in Fig. 7 dargestellt. Prozess 6 wird einer dritten Gruppe von Prozessen zugeordnet.

[0062] 60 In einem vierten Schritt werden die Prozesse P4 und P7 entfernt und einer vierten Gruppe von Prozessen zugeordnet.

[0063] 65 Damit verbleiben lediglich die Prozesse P5 und P8, die in Fig. 8 dargestellt sind. Auf diese verbliebene Prozesse zeigt keine Kante, so dass beide Prozesse einer fünften und letzten Gruppe von Prozessen zugeordnet werden.

[0063] In der Reihenfolge der Schritte oder der Gruppen von Prozessen ist die Reihenfolge gefunden, in der die Prozesse abgearbeitet werden müssen. Die in einem Schritt abgetrennten Prozesse müssen vor den Prozessen ausgeführt werden, die im nächsten Schritt abgetrennt wurden.

[0064] Die Zuordnung der einzelnen Prozesse P1-P8 zu Gruppen von Prozessen G1-G5 ist in Fig. 9 veranschaulicht.

[0065] Da in den Gruppen jeweils mehr als ein Prozess vorhanden sein kann, kann eine Vielzahl von unterschiedlichen, ausführbaren Sequenzen gebildet werden. Aus dieser Vielzahl von gültigen Sequenzen kann beispielsweise diejenige bestimmt werden, die die kürzeste Laufzeit beansprucht oder diejenige, bei der jeweils die Berechnung eines Ausgabewerts und dessen Verwendung als Eingangswert zeitlich am dichtesten aufeinander folgen (Optimierung).

[0066] Fig. 10 zeigt einen Graphen, der eine nicht auflösbare Schleife enthält. Der Graph repräsentiert nicht die Drehmoment-Regelung von Fig. 3.

[0067] Gemäß den oben aufgestellten Berechnungsschritten wurde versucht den Graphen mit dem Werkzeug zum Anordnen oder Prüfen in einzelne Gruppen von Prozessen zu zerlegen. Das Ergebnis der Bearbeitungsschritte 1 bis 3 ist in den Fig. 11 bis 13 dargestellt.

[0068] In Fig. 13, die das Ergebnis des dritten Schritts darstellt, konnte schließlich kein Prozess oder Knoten gefunden werden, zu dem keine Kante des Graphen zeigt. Es gibt also keinen Prozess, der keinen Eingang mit dem Attribut PRE hat. Die verbleibenden Prozesse können somit keiner Gruppe zugeordnet werden. Es liegt eine Schleife vor, die aufgelöst werden muss, um eine ausführbare Task zu erhalten. Die verbleibenden Prozesse werden auf eine Anzeigeeinrichtung oder einen Drucker ausgegeben.

#### Patentansprüche

5

1. Verfahren zum automatischen Gewinnen einer funktionsfähigen Reihenfolge von Prozessen, die aus wenigstens einem Eingabewert einen Ausgabewert bestimmen, mit den Schritten:

40

- eine Vielzahl von Prozessen (P1-P8), deren Eingänge mit wenigstens einem der Attribute
- "Eingangswert des selben Berechnungszyklus" (PRE),
- "Eingangswert des vorhergehenden Berechnungszyklus" (POST),
- "Eingangswert aus beliebigem Berechnungszyklus" (ANY) versehen sind, werden so angeordnet, dass als erster Prozess eines Berechnungszyklus (n) ein Prozess bestimmt wird, der keinen Eingang mit dem Attribut "Eingangswert des selben Berechnungszyklus" (PRE) aufweist,
- die gefundene Reihenfolge der Prozesse (P1-P8) wird ausgegeben.

45

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die einzelnen Prozesse (P1-P8) derart miteinander in Beziehung gesetzt werden, dass zu einem Prozess, dessen Eingang mit dem Attribut "Eingangswert des selben Berechnungszyklus" (PRE) versehen ist, ein Verweis (E) von dem Prozess gesetzt wird, der den Eingangswert liefert, und dass von einem Prozess, dessen Eingang mit dem Attribut "Eingangswert des vorhergehenden Berechnungszyklus" (POST) versehen ist, ein Verweis auf den Prozess gesetzt wird, der den Eingangswert liefert.

60

3. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass für die Berechnung der Reihenfolge der Prozesse (P1-P8) in einem ersten

65

Schritt diejenigen Prozesse entfernt und einer ersten Gruppe von Prozessen (G1) zugeordnet werden, auf die kein Verweis (E) gesetzt ist.

4. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass die verbleibenden Prozesse in weiteren Schritten jeweils auf Prozesse untersucht werden, auf die kein Verweis (E) gesetzt ist, und dass die in jedem Schritt gefundenen Prozesse aus der Menge der zu untersuchenden Prozesse jeweils entfernt und aufeinanderfolgenden Gruppen von Prozessen (G1-G5) zugeordnet werden.

5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass aus einer Lösungsmenge mit berechneten gültigen Reihenfolgen, diejenige Reihenfolge ausgewählt wird, bei der jeweils die Berechnung eines Ausgabewerts und dessen Verwendung als Eingangswert zeitlich am dichtesten aufeinander folgen.

6. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass als letzter Prozess (P1-P8) eines Berechnungszyklus (n) ein Prozess ausgewählt wird, der keinen Eingang mit dem Attribut "Eingangswert des vorhergehenden Berechnungszyklus" (POST) aufweist.

7. Verfahren zum Überprüfen einer Reihenfolge von abzuarbeitenden Prozessen auf Schleifen im Datenfluß, mit den Schritten:

- eine Vielzahl von Prozessen (P1-P8), deren Eingänge mit wenigstens einem der Attribute
- "Eingangswert des selben Berechnungszyklus" (PRE),
- "Eingangswert des vorhergehenden Berechnungszyklus" (POST),
- "Eingangswert aus beliebigem Berechnungszyklus" (ANY) versehen sind, werden so angeordnet, dass in einem ersten Schritt als erster Prozess ein Prozess (P1; P2) bestimmt wird, der keinen Eingang mit dem Attribut "Eingangswert des selben Berechnungszyklus" (PRE) aufweist,
- der erste Prozess (P1; P2) wird einer ersten Gruppe von Prozessen (G1) zugeordnet und aus der Menge der zu untersuchenden Prozesse (P3-P8) entfernt,
- die verbleibenden Prozesse werden in weiteren Schritten jeweils auf Prozesse untersucht, auf die kein Verweis (E) gesetzt ist,
- die in jedem Schritt gefundenen Prozesse werden jeweils aus der Menge der zu untersuchenden Prozesse entfernt und aufeinanderfolgenden Gruppen (G1-G5) zugeordnet, bis in einem Schritt kein Prozess (P1-P8) identifiziert werden kann, der keinen Eingang mit dem Attribut "Eingangswert des selben Berechnungszyklus" (PRE) hat,
- die verbleibenden Prozesse (P1-P8) werden identifiziert und ausgegeben.

8. Werkzeug zum Anordnen oder Prüfen einer Reihenfolge von abzuarbeitenden Prozessen (P1-P8), das aufweist:

- Eingabemittel (2, 21, 3) zur Eingabe einer Vielzahl von Prozessen (P1-P8), deren Eingänge jeweils mit wenigstens einem der Attribute
- "Eingangswerts des selben Berechnungszyklus" (PRE),
- "Eingangswert des vorhergehenden Berechnungszyklus" (POST),
- "Eingangswert aus beliebigem Berechnungszyklus" (ANY) versehen sind,

- wenigstens einen Mikroprozessor (MP) zur Berechnung einer lauffähigen Reihenfolge von Prozesse (P1-P8) nach einem Verfahren gemäß einem der vorhergehenden Verfahrensansprüche,
- ein Ausgabemittel (4) zum Ausgeben der gefundenen Reihenfolge.

---

Hierzu 8 Seite(n) Zeichnungen

---

FIG 1

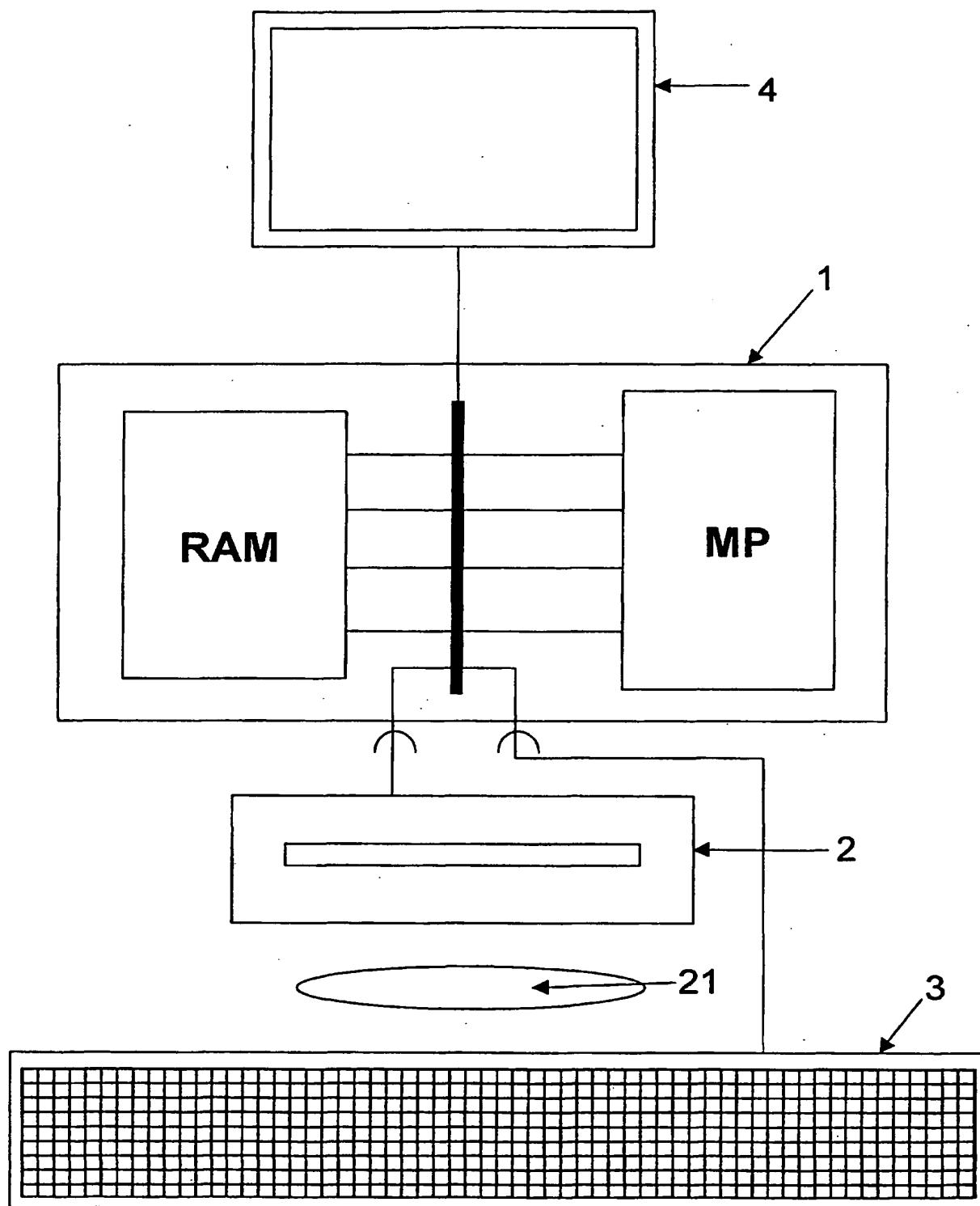


FIG 2

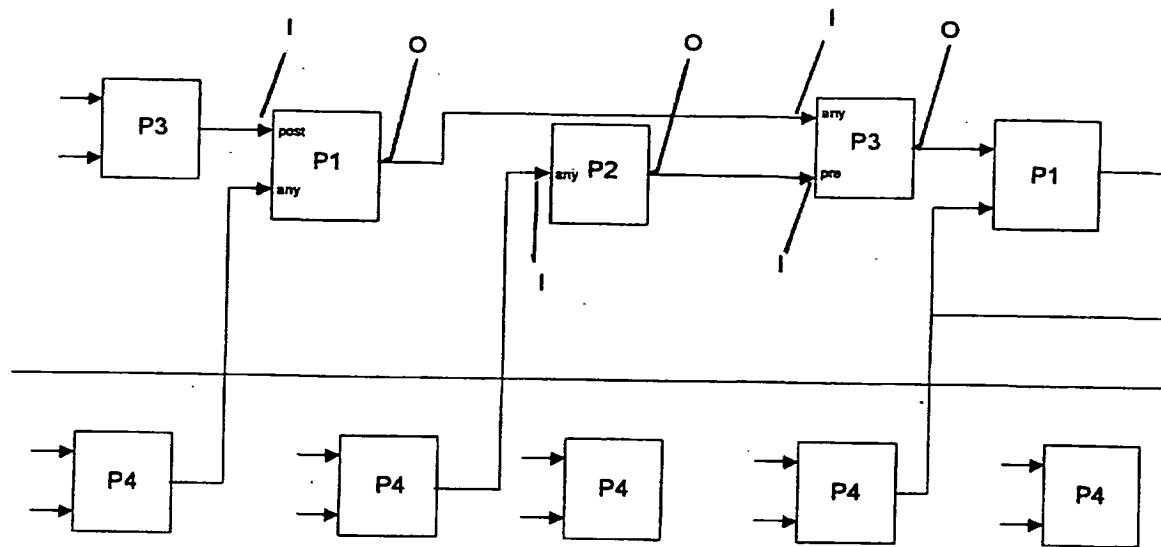


FIG 3

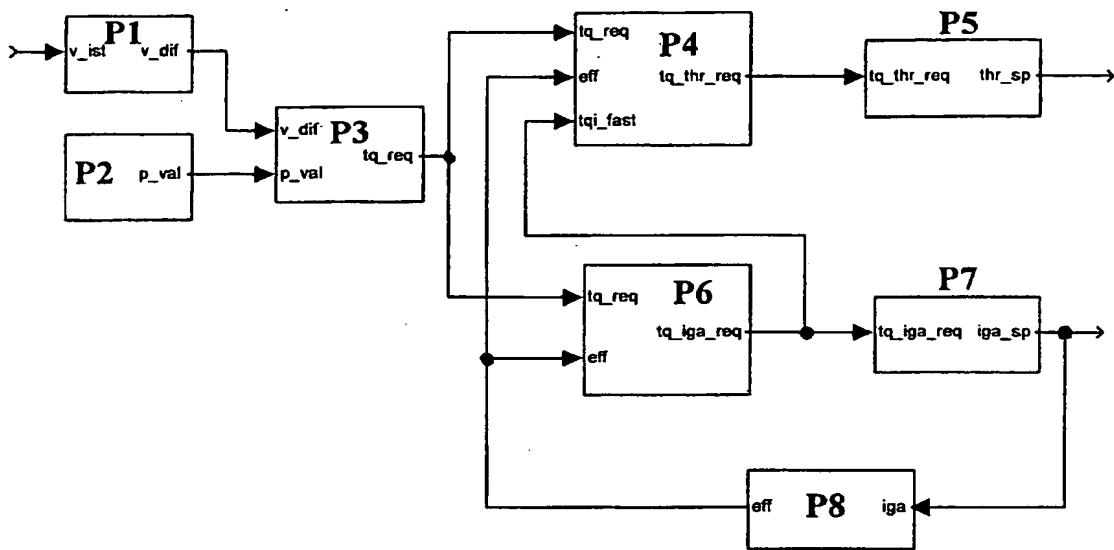


FIG 4

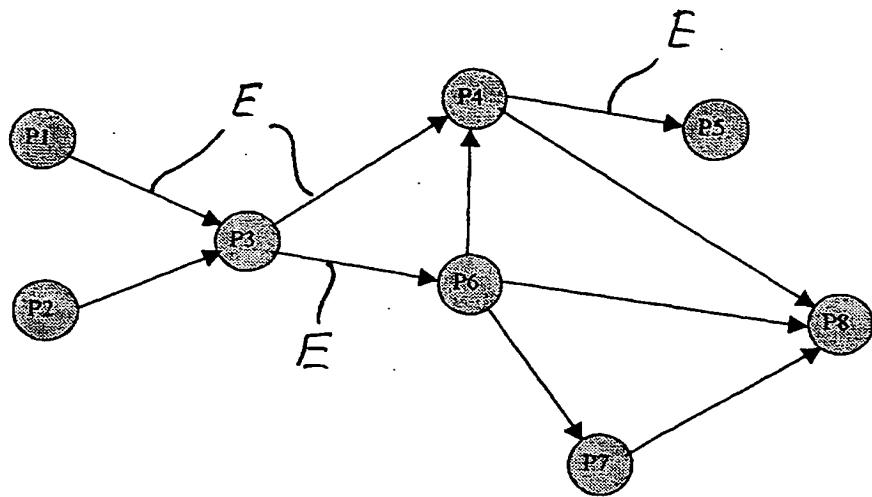


FIG 5

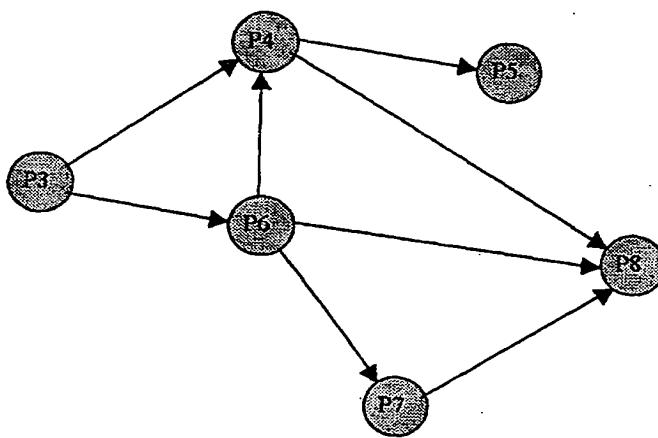


FIG 6

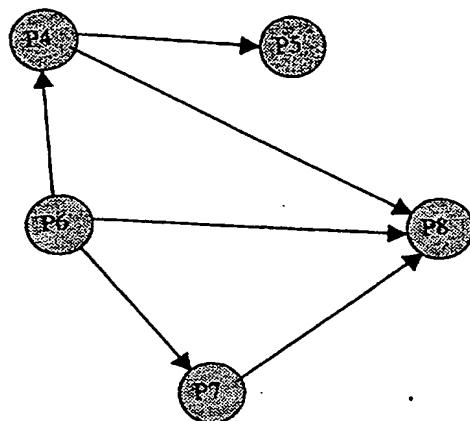


FIG 7

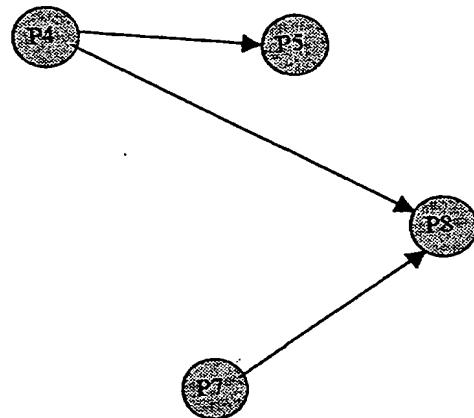


FIG 8

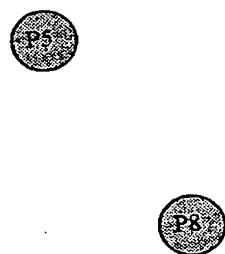


FIG 9

G1	G2	G3	G4	G5
P1	P3	P6	P4	P5
P2			P7	P8

FIG 10

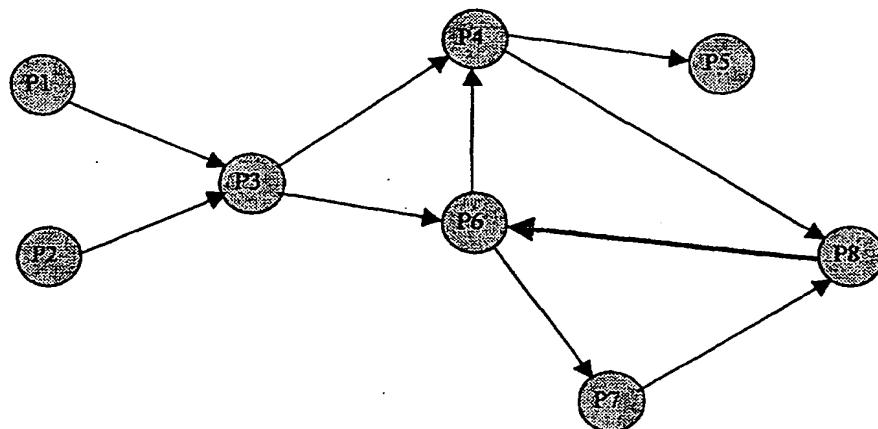


FIG 11

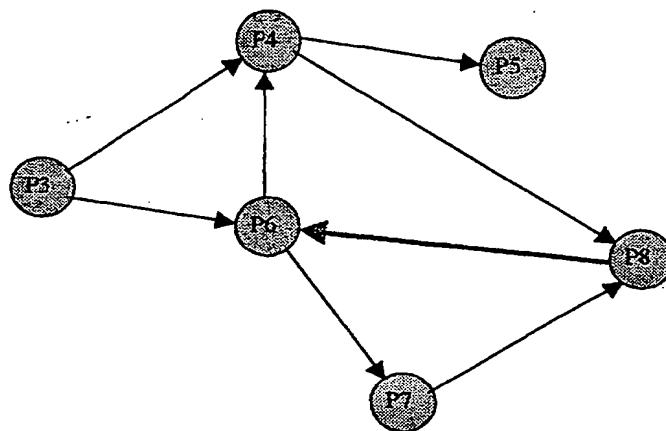


FIG 12

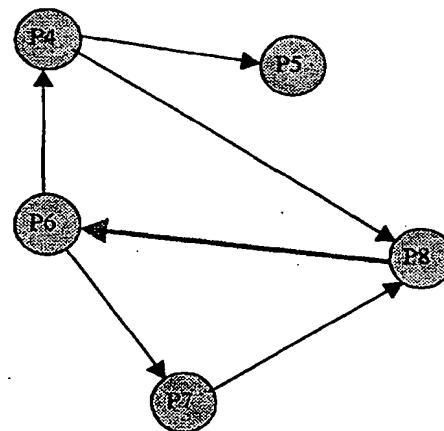


FIG 13

